AUTOMATIC TRANSACTION APPARATUS AND AUTOMATIC TRANSACTION

SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

5   This application is based upon and claims the benefit

of priority from the prior Japanese Patent Application No.

2003-390478, filed on November 20, 2003, the entire contents

of which are incorporated herein by reference.

10   BACKGROUND OF THE INVENTION

1.   Field of the Invention

The present invention relates to an automatic

transaction apparatus and automatic transaction system which

15   execute screen control and automatic transaction operations

according to the screen content from the Web server, and more

particularly to an automatic transaction apparatus and

automatic transaction system which operates by the screen

content, where the screen information and apparatus control

20   information related to the screen are embedded.

2.   Description of the Related Art

Automatic transaction apparatuss are used for various

transactions, and in the financial field, for example,

25   automatic withdrawal machines and automatic

deposit/withdrawal machines are used, and in other fields,

automatic ticket machines and automatic issuing machines are

1

used. In such automatic transaction apparatuss, automatic

transaction apparatuss for depositing/withdrawing, issuing

tickets and outputting various information using Web

technology are provided as such networks as the Internet

5   currently develop.

Fig. 21 is a block diagram of a conventional automatic

transaction system, and shows an ATM (Automatic Teller

Machine) for financial operations. As Fig. 21 shows, the WWW

(World Wide Web) server 300 and the automatic cash

10   transaction apparatus 400 are connected via a network.

According to the request of the automatic transaction

apparatus (ATM) 400, which is a client, the server 300

transmits the Web page (screen content) 500 to the ATM 400.

This Web page is created with a program for creating a screen

15   to be displayed on the display device using a page

description language (HTML, JAVA (registered trademark)

script), and relating to the display content of this one page

(one screen), the control program of another device (e.g.

card processing device, cash processing device, pass book

20   processing device, itemized slip processing device), for

which drive is controlled, is embedded as an object.

For example, as Fig. 21 shows, the Web page 500 is

comprised of a screen creation program, applet tag for

specifying an object (applet), screen content 502 for

25   specifying the execution method (method) by script, and

applet 510 which sets the program for executing the method of

the object (applet), using the page description language HTML

2

(Hyper Text Markup Language.

This Web page 500 is downloaded from the WWW server 300 to the browser 410 of the ATM 400. In the ATM 400, on the other hand, the ATM middleware operates under the control of a kernel (OS), and performs I/O operations (transaction operations). The ATM 400 has a card reader/writer unit 440, receipt/journal printer 441, bill/coin processing unit 442, passbook processing unit 443 and customer operation panel as the I/O mechanical units.

According to the screen creation program of the Web page, the browser 410 displays the screen on the customer operation panel, analyzes the applet tag of the screen content 502 and the method name, executes the corresponding program of the applet 510, and issues commands to the I/O units 440 - 443.

In such ATM control by a Web browser, it is proposed to specify the operation method (initialization in this example) of each device using the script (Java script) of the screen content 502 by specifying the device interface sorting section (machine ID) as an embedded object (applet) of the screen content 502 for each device (e.g. cash processing device) as shown in Fig. 21 and Fig. 22 (e.g. Japanese Patent Application Laid-Open No. 2000-298752).

In this method, the device interface sorting section 420 is specified by the applet tag of the screen content 502, and the device sorting section 420 sequentially reads the script, decodes it, sorts it to the interface sections 430 - 433 which handles the operation instructions, and operates the

3

corresponding I/O units 440 - 443.

This example shows the case of the device initialization operation command, so as Fig. 23 shows, the Initialize ("cash") Script in the screen content is read, the

5   initialization command is sent to the controller of the cash processing device 442, the initialization completion reply is received, then the Initialize ("card") Script is read, the initialization command is sent to the controller of the card processing device 440, and the initialization completion

10   reply is received. Hereafter Initialize ( ) Script is sequentially read, the initialization command is sent to the controller of the corresponding processing device, the initialization completion reply is received, and processing ends.

15   In this proposal, when ATMs with the same functions are controlled via the Web, a plurality of units can be operated by one applet tag of the screen content, the description of the HTML of the screen content can be short, and the embedded object can be simple.

20   However ATMs do not always have the same function, for there are many ATMs which have different functions. For example, cash processing functions are processing bills and coins or processing only bills, or some ATMs execute pass book processing while others do not, or some ATMs execute

25   deposit/withdrawal processing and other execute only withdrawal.

In order to Web-control such ATMs with different

4

functions, the script itself of the screen content must be changed considerably, even with a same applet tag, according to the constituting functions and units, since the method to be called up is for each unit in the case of the prior art.

5   In other words, the description of the screen content must be designed according to the differences of the constituting functions and units of the ATM to be controlled.

Therefore it takes enormous effort to develop a Web page on a WWW server 300, and it takes time to add new functions
10   (e.g. link function with a portable telephone).

In the prior art, a method is called up for each unit, so the script of the screen content must be created for each unit. Therefore, as the above example of initialization shown in Fig. 23 shows, the command is issued and the reply
15   is received for each script in order to control the synchronization of the plurality of units, therefore the synchronization control takes time, and the time for a user to use the automatic transaction apparatus becomes lengthy, which drops the operation rate, and increases the reply wait
20   time.

SUMMARY OF THE INVENTION

With the foregoing in view, it is an object of the
25   present invention to provide an automatic transaction apparatus and automatic transaction system for controlling a plurality of automatic transaction apparatus which have

different functions with decreasing the change of description of objects in the screen content.

It is another object of the present invention to provide an automatic transaction apparatus and an automatic transaction system for simplifying the creation of a Web page of the Web server when a plurality of automatic transaction apparatus which have different functions are controlled.

It is still another object of the present invention to provide an automatic transaction apparatus and an automatic transaction system for improving the speed of synchronization control of a plurality of different units, even using Web control, and decreasing the reply wait time.

To achieve these objects, the automatic transaction apparatus of the present invention is an automatic transaction apparatus for communicating with a Web server and performing guide display and transaction operation according to the operation of the user, having a display unit for performing the guide display, a plurality of I/O units for performing the transaction operation, and a control unit for controlling the guide display of the screen of the display unit according to the screen content from the Web server, and controlling the plurality of I/O units according to the objects embedded in the screen content. And the control unit calls up a method for each processing of the transaction operation for controlling the synchronization of the plurality of I/O units by the script of the object, and controls the synchronization of the plurality of I/O units.

6

The automatic transaction system of the present

invention has a Web server, and an automatic transaction

apparatus which is connected to the Web server via a network

for communicating with the Web server and performing guide

5   display and transaction operation according to the operation

of a user. And the automatic transaction apparatus has a

display unit for performing the guide display, a plurality of

I/O units for performing the transaction operation, and a

control unit for controlling the guide display of the screen

10   of the display unit according to the screen content from the

Web server, and controlling the plurality of I/O units

according to the objects embedded in the screen content. And

the control unit calls up the method for each processing of

the transaction operation for controlling the synchronization

15   of the plurality of I/O units by the script of the object,

and controls the synchronization of the plurality of I/O

units.

In the present invention, it is preferable that the

control unit has a browser which interprets the screen

20   content from the Web server, and performs the guide display,

and also interprets the script of the object embedded in the

screen content, and calls up the method for each processing

of the transaction operation for controlling the

synchronization of the plurality of I/O units, and controls

25   the synchronization of the plurality of I/O units from the

browser.

In the present invention, it is preferable that the

7

control unit transmits the request to the Web server

according to the post request by the called up method.

In the present invention, it is preferable that the

control unit receives the screen content which has the screen

creation program described by a page description language,

the script of the object, and the method program called up by

the script.

In the present invention, it is preferable that the

control unit issues the operation command to the plurality of

I/O units by the called up method, and receives the reply

from the I/O units.

In the present invention, it is preferable that the

plurality of I/O units has at least a cash processing unit

and a card processing unit.

In the present invention, it is preferable that the

control unit specifies the plurality of I/O units for which

synchronization is controlled by the method according to the

input parameters attached to the script.

In the present invention, it is preferable that the

browser creates the guide screen by the screen creation

program described by the page description language of the

screen content, calls up the method program from the script

of the object, and controls the synchronization of the

plurality of I/O units.

In the present invention, it is preferable that the

browser creates the guide screen by the screen creation

program described by the page description language of the

screen content, calls up the method program of an applet from the applet specifications and the method specifications of the script of the object, and controls the synchronization of the plurality of I/O units.

5   In the present invention, it is preferable that the browser issues an operation command to a plurality of I/O controllers for controlling each of the I/O units by the called up method, and receives replies from the I/O controllers.

10   According to the present invention, objects for each processing of the transaction processing are embedded in the screen content of the Web server, and one method for each processing is called up, so the operation of a plurality of I/O units can be controlled, and the I/O units can be

15   commonly used for different automatic transaction apparatus with a general transaction flow. Therefore screen content for Web-controlling automatic transaction apparatus with different functions and configurations can be easily created.

       Also one method is called up, so parallel control is

20   possible and high-speed I/O unit control can be implemented. By this, the wait time of a user for the automatic transaction apparatus can be decreased, and the operation rate can be improved.

25   BRIEF DESCRIPTION OF THE DRAWINGS

       Fig. 1 is an external view of the automatic transaction

apparatus according to an embodiment of the present

invention;

Fig. 2 is a block diagram depicting the automatic

transaction apparatus in Fig. 1;

5   Fig. 3 is a system block diagram depicting the automatic

transaction system according to an embodiment of the present

invention;

Fig. 4 is a diagram depicting the browser and screen

content in Fig. 3;

10   Fig. 5 is a block diagram depicting the ATM middleware

in Fig. 3 and Fig. 4;

Fig. 6 is a table showing the transaction commands of

the common interface in Fig. 3 and Fig. 5;

Fig. 7 is a table describing the agents in Fig. 3 and

15   Fig. 4;

Fig. 8 is a table describing other agents in Fig. 3 and

Fig. 4;

Fig. 9 shows the screen content in Fig. 3 and Fig. 4;

Fig. 10 is a diagram depicting the I/O control operation

20   by the agent in Fig. 3 and Fig. 4;

Fig. 11 is a diagram depicting the input parameter of

the screen content in Fig. 9;

Fig. 12 is a diagram depicting the operation of the

method of the agent in Fig. 10;

25   Fig. 13 is a diagram depicting deposit processing

according to an embodiment of the present invention;

Fig. 14 is a diagram depicting the middleware API

conversion operation according to an embodiment of the present invention;

Fig. 15 is a diagram depicting the middleware API conversion operation according to another embodiment of the present invention;

Fig. 16 is a table showing common interfaces and vender specific interfaces according to an embodiment of the present invention;

Fig. 17 is a flow chart depicting initialization interface conversion processing in the I/O control library;

Fig. 18 is a flow chart depicting interface conversion processing of the pass book insertion processing in the I/O control library;

Fig. 19 is a table describing agents according to another embodiment of the present invention;

Fig. 20 shows screen content according to another embodiment of the present invention;

Fig. 21 is a diagram depicting the automatic transaction system by conventional Web control;

Fig. 22 shows screen content of conventional Web control; and

Fig. 23 is a diagram depicting the sequence of the I/O control operation by the script of conventional screen content.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention will now be
described in the sequence of automatic transaction system,
I/O control mechanism by Web control, automatic transaction
processing, customization method for conventional middleware,
5 and other embodiments.

[Automatic transaction system]

Fig. 1 shows an external view of an automatic
transaction apparatus according to an embodiment of the
present invention, Fig. 2 is a block diagram depicting the
10 automatic transaction apparatus in Fig. 1, and Fig. 3 is a
system block diagram depicting the automatic transaction
system according to an embodiment of the present invention.

As Fig. 1 shows, the automatic transaction apparatus 1
has a card entry 2 for inserting and ejecting a magnetic card,
15 a pass book entry 3 for inserting and ejecting a magnetic
pass book, a bill entry 4 for entering and ejecting bills, a
coin entry 5 for entering and ejecting coins, a UOP (User
Operation Panel) 6 for a user to operate, an operation
display 7 for displaying the operation status to the user,
20 and a customer sensor 8 for detecting the user.

Fig. 2 is a block diagram depicting the automatic
transaction apparatus 1 in Fig. 1. A CRW (Card Reader Writer)
unit 10 reads the magnetic card by the magnetic head, and
returns it to the entry 2, while transporting the magnetic
25 card inserted from the card entry (card insertion slot) 2
using a transport mechanism, which is not illustrated. The
CRW unit 10 has an image sensor so as to read the magnetic

12

card (embossed section) optically.

The RPR (Receipt Printer) unit 20 prints the transaction result on the receipt paper by the print head, and ejects the receipt to the card entry 2. The RPR unit 20 stores the

5 receipt returned from the entry 2 when the ejected receipt is not removed by the user within a predetermined time.

The JPR (Journal Printer) unit 70 prints out the transaction status and the result on the journal printer by the print head. The UOP (User Operation Panel) unit 30 is

10 comprised of the UOP (display with touch panel) 6 and the control circuit thereof. The pass book (PPR) processing unit 40 reads the magnetic pass book inserted from the pass book entry 3, prints the transaction on the magnetic pass book, and ejects the pass book from the pass book entry 3.

15 The bill/coin processing unit 50 performs the deposit operation by validating the bills and the coins entered through the bill entry 4 and the coin entry 5, counting them and storing them in stackers, and performs withdrawal operation for taking off the requested bills and coins from

20 the cash stacker, and releasing them to the bill entry 4 and the coin entry 5.

The control section 60 is connected to these control units 10, 20, 30, 40, 50 and 70 via such a network 90 as a LAN, and performs automatic transaction processing based on

25 the software configuration, which is mentioned later in Fig. 3.

Fig. 3 is a block diagram of the automatic transaction

system according to an embodiment of the present invention.
The automatic transaction apparatus 1 exchanges commands,
parameters and data required for the transaction processing
with the WWW (World Wide Web) server (host) 100 via such a

5   network 110 as the Internet.

In the automatic transaction apparatus 1, the mentioned
control section 60 has the browser 120, ATM middleware 130,
kernel (OS) 140 and device driver 150.

The device driver 150 is comprised of a card unit driver

10   151 for driving the card (CRW) unit 10, a receipt/journal
unit driver 152 for driving the receipt/journal units
(RPR/JPR) 20 and 70, a BRU driver 153 for driving the BRU
(bill) unit 50, a CRU driver 154 for driving the CRU (coin)
unit 50, a graphic driver 155 for driving the UOP 30, a touch

15   screen driver 156, and a PPR driver 157 for driving the pass
book unit 40.

The browser 120, which is a Web browser, such as
Internet Explorer (Microsoft trademark), requests the Web
server 100 to transmit content, and interprets and displays

20   the screen content (Web page) transmitted by the Web server
100.   In this case, the browser 120 requests the Web page
required for the transaction processing, which is constructed
by HTML and Java script, interprets the transmitted Web page,
and controls the screen of the ATM middleware 130 and the UOP

25   30.

The kernel 140 is a known OS (Operating System), such as
Windows (registered trademark) and Linux, and under the

operating environment of the kernel 140, the browser 120, ATM

middleware 130 and device driver 150 operate.

The ATM middleware 130 is comprised of a parameter file

160, I/O control layer 170, I/O client layer 190, I/O server

5   layer 200 and I/O service provider layer 210.

This I/O client layer 190 is for controlling an

individual I/O unit installed in the apparatus, where the I/O

server layer 200 starts up and ends an I/O operation and

controls the communication protocol, and the I/O service

10   provider layer 210 converts the messages for each I/O unit.

These are conventional middleware 180, which were designed

according to the functional range and the type of the

apparatus and to the specifications of the connected I/O unit.

The I/O control layer 170, on the other hand,

15   transmits/receives commands and data using the common

application interface protocol of the Web server 100 and the

middleware. The functional ranges of the apparatus are

different from each other, depending on the model of the

apparatus, so the common application interface (API) is

20   comprised of common commands and data systems that can

operate all models, which will be described later in Fig. 6.

The I/O control layer 170 integrates the application

interface (API) of the I/O client layer 190, and constructs a

more abstract common API. The parameter file 160 is for

25   storing the input parameters/fixed parameters which are

uniquely determined by the system specifications specific to

the vender (ATM manufacturer).

15

The I/O control layer 170 calls up parameters specific

to each client layer from this parameter file 160 when

calling up the I/O client layer 190, and converts the common

API into the conventional client API.

5   By this, the highly abstract common API can be converted

into a client API, matching the ATM middleware 190 of the

automatic transaction apparatus 1 and the type of the

installed I/O units, and the conventional ATM middleware 180

and the I/O unit can be operated. In other words, a

10   conventional ATM middleware can be customized so as to

operate with a common API.

As Fig. 3 shows, in the present invention, the agent 123

defined for each processing of the ATM is embedded as an

applet of the screen content 122 of the Web page, and

15   operation of the I/O units required for processing are

controlled for each processing. Details will be described in

Fig. 4 and later, but the applet name is defined for each

processing, such as synchronization control and

initialization control of synchronization control (hereafter

20   called Agent), and the method is also provided for each

processing. In other words, an applet (Agent) having a

method for controlling the operation of the I/O unit

corresponding to the processing is provided, and the applet

(Agent) and the method are specified for each processing.

25   By this, even if a plurality of units operate

synchronously, the operation can be executed for each

processing, and the applet name can be assigned for each

processing, so the changed locations of a Web page can be decreased in the Web control of ATMs with different functions, and a Web page can be easily created even in complicated automatic transaction control.

5   Since a plurality of I/O units can be controlled merely by calling up one method, a plurality of I/O units can be controlled in parallel, processing speed can be increased accordingly, the ATM control time can be decreased even if Web control is used, and the wait time for a user can be

10   decreased.

[I/O control mechanism by Web control]

Now the I/O control mechanism by Web control will be described. Fig. 4 is a block diagram depicting the Web browser and the I/O controller in the configuration in Fig. 3,

15   Fig. 5 is a block diagram depicting the ATM middleware in the configuration in Fig. 3, and Fig. 6 is a table showing the common API commands in Fig. 3 and Fig. 4.

As Fig. 4 shows, the screen content (HTML + Java script) 122 and the Agent (applet) 123 are downloaded from the WWW

20   server 100 to the Web browser 120. The Agent 123 is comprised of the control Agent group 124 for controlling each unit, POST Agent 126 for performing post processing, View Agent 128 for constructing the screen, and communication DLL (Dynamic Link Library) 129 for communicating with the I/O

25   controller 170 via the Java Native Interface and the common API interface 132.

If the interfaces are the same, then the communication

17

DLL group 129 is unnecessary. For the control Agent group

124, only the control agents required for the display screen

are downloaded. As described in detail in Fig. 7 and later,

the control agent is constructed for each ATM transaction

5 processing.

In the operation of the configuration in Fig. 4, the

method of the control Agent 124 is called up by the method

name described in the Java script of the screen content 122.

The called up method issues a command to the I/O controller

10 170 via the communication DLL group 129.

The I/O controller 170 operates the corresponding unit

via the ATM middleware 180, and receives the operation

completion notice. The I/O controller 170 replies the

command execution result to the control Agent 124. The

15 control Agent 124 sends the post request or drawing request

to the POST Agent 126 for executing post processing or to the

View Agent 128 for updating the screen, and has either POST

processing or screen update processing executed.

Before describing this Agent, the common API will be

20 described first. Fig. 6 shows an example of the command

types of the common API. As CRW (Card Reader/Writer)

commands, a card insertion command and a card ejection

command are provided.

As RPR (Receipt Printer) commands, a printing command,

25 release command and other commands are provided. As PPR

(Pass book Printer) commands, a pass book insertion command,

printing command, MS (Magnetic Stripe) write command, pass

book ejection command, auto turn page command and other commands are provided.

As BRU (Bill Recycle Unit) commands, an initialization command, acceptance/counting command, storing command,

5 deposit return command, feeding command, release command, capturing command, transport path check command, jam reset command and other commands are provided. The CRU (Coin Recycle Unit) commands are the same as the BRU commands, for which description is omitted.

10 The configuration of the ATM middleware 130 in Fig. 3 and Fig. 4 will now be described with reference to Fig. 5. The I/O control layer 170 has the I/O control library group (I/O controllers) 171 - 178 for controlling each I/O.

In this case, the I/O control library group is comprised

15 of the pass book control library 171, CRW control library 172, ten key control library 173, receipt control library 174, bill control library 175, coin control library 176, journal control library 177 and transaction control library 178.

These control libraries 171 - 178 are called up by a

20 task (e.g. card control EXE) specified by the common API, and converts the task into the client API of conventional middleware using the above mentioned parameter table 160.

The I/O client layer 190 of the conventional middleware 180 is for controlling an individual I/O unit installed in

25 the apparatus, and in this case, the card (CRW) client 191, coin client 192, bill client 193, RPR client 194, JPR client 195 and PPR client 196 are provided.

In the same way, the I/O server layer 200 is also divided into individual I/O units for starting and ending an individual I/O operation, and controlling the communication protocol. In other words, the card (CRW) server 201, coin server 203, bill server 202, RPR (Receipt Printer) server 204, JPR server 205 and PPR (pass book printer) server 206 are provided.

In the same way, the I/O service provider layer 210 is also divided into individual I/O units for converting the messages for each I/O unit. In other words, the card (CRW) service provider 211, coin service provider 213, bill service provider 212, RPR service provider 214, JPR service provider 215 and PPR service provider 216 are provided.

In other words, the control library, the client, the server and the service provider which constitute the ATM middleware are provided corresponding to each I/O unit, and the I/O control layer 170 converts the requested commands and the parameters of the common API into the commands and the parameters of the conventional middleware API, and operates the I/O unit via the conventional middleware.

Now the above mentioned Agent will be described with reference to Fig. 7 and Fig. 8. As Fig. 7 shows, the synchronization Agent is a program for controlling the synchronization of a plurality of I/O units, and has such methods (programs) as initialization, mechanism reset, bill/coin insertion, medium simultaneous ejection, printing/feeding/MS write/ejection preparation, deposit

return, storing, forced ejection/capturing, unit information
acquisition/transaction status setting/two-screen display
control, deposit/withdrawal preparation, forced replenishment,
jam reset and card/pass book insertion.

5   Each method issues a command to the I/O controller
indicated by a black dot in Fig. 7, and receives a reply on
the command execution result. For example, the
initialization method issues the initialization command to
the bill controller 175, coin controller 176, pass book

10   controller 171, card controller 172, receipt controller 174,
journal controller 177, transaction controller 178 and ten
key controller 173 (see Fig. 5), has each controller execute
initialization processing, and receives the initialization
processing result from each controller as the reply.

15   In the same way, the mechanism reset method issues the
mechanism reset command to the bill controller 175, coin
controller 176, pass book controller 171, card controller 172,
receipt controller 174 and the journal controller 177 (see
Fig. 5), has each controller execute mechanism reset

20   processing, and receives the mechanism reset processing
result from each controller as the reply.

In the same way, the bill/coin insertion method issues
the insertion command to the bill controller 175 and coin
controller 176 (see Fig. 5), has each controller execute

25   insertion processing, and receives the insertion processing
result from each controller as the reply.

In the same way, as the POST Agent, the POST processing

21

method and the POST data holding method are provided. Also as the View (text display) Agent, the font setting method, text display method and text erasing method are provided.

Also as Fig. 8 shows, the applet for controlling a

5 single I/O unit as well, is defined as an Agent for being controlled in a same architecture. In other words, as Fig. 8 shows, the bill control Agent, pass book control Agent, card control Agent, receipt control Agent, transaction control Agent and ten key Agent are provided.

10 The bill control Agent has an acceptance/counting method, storing method, deposit return method, releasing method and cancellation method for controlling each bill controller 175. The pass book control Agent has a line set/page mark read method, MS (Magnetic Stripe) read method, auto turn page

15 method, page check auto turn method and pass book configuration information setting method for controlling each pass book controller 171.

The card control Agent has a card insertion method, cancellation method, money transfer card printing method,

20 money transfer card issuing method and ejection preparation method for controlling each card (CRW) controller 172. The receipt control Agent has an overlay registration method and ejection preparation method for controlling each receipt controller 174.

25 In the same way, the transaction control Agent has a transaction information setting method, apparatus status monitoring method, apparatus status acquisition method,

22

operation information setting method and cancellation method for controlling each transaction controller 178. The ten key Agent has a ten key input start method and ten key input end method for controlling each ten key controller 173.

5   The Web operation to call up these Agents and methods will be described with reference to Fig. 9 to Fig. 12. Fig. 9 shows the screen content to be transmitted by the Web server 100, and Fig. 10 is a diagram depicting the I/O control by the screen content in Fig. 9.

10   As Fig. 9 shows, the Agent name to be called up in the screen is specified in the screen content described with HTML (page description language). As an example, the synchronization control (Sync) Agent is called up here by the applet tag <APPLET CODE = "U_agtSync.class".

15   Also by the description <SCRIPT Language = "javaScript>, the call-up of the method of the synchronization control Agent is specified using Java script. In other words, the call-up of the initialization method of the synchronization control Agent is specified by the description ret =

20   document.U_agtSync.initial. Here the details of the screen display content are omitted.

When this screen content is downloaded to the browser 120, the synchronization Agent is specified by the applet tag, as shown in Fig. 10, and the initialization method is called

25 up by the call-up method name of the SCRIPT.

As described above, the called up initialization method issues the initialization command to the bill controller 175,

coin controller 176, pass book controller 171, card

controller 172, receipt controller 174, journal controller

177, transaction controller 178 and ten key controller 173

(see Fig. 5), has each controller execute initialization

5   processing, and receives the initialization processing result

from each controller as the reply.

In this case, as Fig. 12 shows, the called up

initialization method continuously issues the initialization

command to the bill controller 175, coin controller 176, pass

10   book controller 171, card controller 172, receipt controller

174, journal controller 177, transaction controller 178 and

ten key controller (see Fig. 5), and receives the replies

from these controllers sequentially as the processing

completes.

15   Therefore the I/O units can be controlled in parallel,

and even a plurality of I/O units can be controlled in a

short time. Whereas in the case of the method of specifying

a method for each I/O unit using conventional Script, which

is a sequential control for issuing commands and receiving

20   replies sequentially for each Script as shown in Fig. 23, it

takes time to control the I/O by the Web.

Also the above mentioned Agent is provided for each

processing to which a processing unit name is assigned, so

ATMs with various functions can be supported with the same

25   applet tags and method names merely by slightly changing the

content of this processing unit. In this example, the

content of the processing unit is changed by changing the

24

input Param in the parenthesis of the method name of the

above mentioned method call-up SCRIPT.   In other words, as

Fig. 11 shows, the input Param is comprised of several bytes

of bit information (8-bit information in Fig. 11) for setting

5 the input flag for each I/O controller (bill, coin, pass book,

card, receipt, journal, transaction control and ten-key).

The input flag indicates that the input of the method to

the I/O controller where "1" is set is enabled.   The called

up method refers to this input flag, and determines the I/O

10 controller to which the command is issued.   Therefore even

ATMs with different functions can execute a same processing

with a same applet tag and method name by operating the input

flag in the Web server 100.

For example, if the input flag of coins is set to "0"

15 for the ATM without handling coin, issuing a command to the

coin controller can be prevented.   In the same way, if the

pass book input flag is set to "0" in a transaction apparatus

which does not handle pass books, issuing a command to the

pass book controller can be prevented.

20 The POST Agent is specified within this agent by

ParseInt (postMode) in the parenthesis of the method name of

the SCRIPT.   Therefore POST processing can be smoothly

executed by the agent.   The data can be posted at the timing

when the agent received all the command completions.

25 [Automatic transaction processing]

Now the relationship of the screen content of the Web

server 100, customer operation screen (UOP screen), Agent/

25

method, and I/O controller will be described using the card deposit transaction in Fig. 13 as an example.

The Web server 100 issues the transaction type select screen content and applet to the ATM 1 by JSP (Java Server
5 Pages). In the ATM 1, the browser 120 displays the transaction type select screen on the customer operation screen of the UOP 6.

And the deposit/withdrawal preparation method of the synchronization control Agent in Fig. 7 is called up by the
10 applet name and method name embedded in the screen content, the deposit/withdrawal preparation command is issued to the bill controller 175 and the coin controller 176 to prepare for deposit/withdrawal. When deposit/withdrawal preparation completes, both controllers 175 and 176 reply the completion
15 to the deposit/withdrawal preparation method.

If the deposit key is pressed in the type select screen of the UOP 6, this is reported to the Web server 100. By this, the Web server 100 moves to the card start processing by JSP, and issues the screen content of the insertion
20 processing to the ATM 1. In the ATM 1, the browser 120 displays the card insertion screen on the customer operation screen of the UOP 6.

Then the card insertion method of the card control Agent is called up by the applet name card control, which is
25 embedded in the screen content, and the name card insertion, which is a method name, and the card insertion command is issued to the card controller 172. The controller 172

26

operates the card unit 10 by the card insertion command. When the card unit 10 detects the insertion of the card and reads the card, the card control library 172 replies the completion of the card insertion to the card insertion method.

5 And the card insertion method requests to post the transaction status (in this case, card insertion detection and card read data), and the POST Agent 126 sends the request to the Web server 100. In this case, the synchronization Agent card/pass book insertion method is called up, and this

10 input flag (see Fig. 11) is set at the card controller, then the same control is possible.

Then the JSP of the Web server 100 issues the screen content of the cash insertion processing to the ATM 1 by the request at the end of the card insertion processing. In the

15 ATM 1, the browser 120 displays the cash insertion screen on the customer operation screen of the UOP 6 according to the screen content. And the bill/coin insertion method of the synchronization control Agent is called up by the card control Agent, which is an applet name embedded in the screen

20 content, and by the card insertion, which is a method name, and the acceptance/counting command is issued to the bill controller 175 and the coin controller 176.

The bill controller 175 and the coin controller 176 operate the bill unit and the coin unit 50 by the

25 acceptance/counting command. The bill and coin units 50 opens the insertion port, detects the inserted bills and coins, closes the insertion port, and counts the inserted

27

bills and coins. When counting ends, the bill controller 175

and the coin controller 176 replies the completion of

acceptance/counting to the bill/coin insertion method. And

the bill/coin insertion method requests to post the

5   transaction status (number of accepted bills/coins and total

amount), and the POST Agent 126 sends the request to the Web

server 100.

Then the JSP of the Web server 100 moves to deposit

processing, such as a balance update, by the request at the

10   end of cash insertion processing, issues the screen content

of the computer processing to the ATM 1, and displays the

"Please Wait" screen on the customer operation screen of the

UOP 6.

And the browser 120 calls up the printing/feeding/MS

15   write/ejection preparation method (see Fig. 7) by the

synchronization control Agent which is an applet name

embedded in the screen content, and printing/feeding/MS

write/ejection preparation which is a method name, and issues

the MS write command and ejection preparation command to the

20   card controller 172, the printing command and ejection

preparation command to the receipt controller 174, and the

printing command to the journal controller 177.

By this, the card unit 10 writes the magnetic stripe of

the card, and prepares for ejection of the card, the receipt

25   printer 20 prints the receipt and prepares for ejection, and

the journal printer 70 prints the journal. Each controller

172, 174 and 177 replies completion to the

printing/feeding/MS write/ejection preparation method by the completion of command execution.

And the printing/feeding/MS write/ejection preparation method requests to post the transaction status (ejection
5 preparation completion in this case), and the POST Agent 126 sends the request to the Web server 100.

The JSP of the Web server 100 moves to the medium ejection processing and issues the screen content of the medium ejection to the ATM 1. In the ATM 1, the browser 120
10 displays the medium ejection screen on the customer operation screen of the UOP 6. And the browser 120 calls up the medium ejection method (see Fig. 7) of the synchronization control Agent by the synchronization control Agent, which is an applet name embedded in the screen content, and medium
15 ejection, which is a method name, and issues the releasing command to the card controller 172, releasing command to the receipt controller 174, and bell regeneration command to the sound controller, which is not illustrated.

By this, the card unit 10 releases the card, and the
20 receipt printer 20 releases the receipt, removal of the card and receipt is detected, and if the card and receipt are not removed within a predetermined time, a continuous tone is sounded. Each controller 172 and 174 replies completion to the medium ejection method at the completion of the command
25 execution.

And the medium ejection method requests to post the transaction status (removal completion in this case), and the

POST Agent 126 sends the request to the Web server 100.

Then the JSP of the Web server 100 receives the request at the removal completion from the ATM 1, moves to the transaction end processing, and issues the screen content of 5 the transaction end to the ATM 1. In the ATM 1, the browser 120 displays the end screen on the customer operation screen of the UOP 6. The browser 120 calls up the bill/coin method (see Fig. 7) of the synchronization control Agent by the synchronization control Agent, which is an applet name 10 embedded in the screen content, and by the bill/coin storing, which is a method name, and issues the storing command to the bill controller 175 and the coin controller 176.

By this, the bill/coin unit 50 stores the counted bills and coins in the internal stacker. Each controller 175 and 15 176 replies completion to the bill/coin method at the completion of the command execution. And the bill/coin method requests to post the transaction status (storing completion in this case), and the POST Agent 126 sends the request to the Web server 100.

20 By this, the Web server 100 returns to the above mentioned transaction type select screen, and repeats the same processing.

In this way, the Agent for each processing of the transaction processing is embedded in the screen content of 25 the Web server 100, and one method, which is a processing unit, is called up, so the operation of a plurality of I/O units can be controlled, and different automatic transaction

apparatus can be used in common in a general transaction processing flow. Therefore the screen content for controlling automatic transaction apparatus having different functions and configurations by the Web can be easily created,

5   parallel control is possible, and high-speed I/O unit control can be implemented. By this, the wait time of a user for the automatic transaction apparatus can be decreased, and the operation rate can be improved.

In this example, the card deposit processing was used

10   for description, but this processing is the same for withdrawal processing, deposit processing/withdrawal processing using a pass book, account printing processing and balance inquiry.

[Customization method of conventional ATM middleware]

15   Now a method of operation with a common interface, even if the specifications of an I/O unit and the specifications of ATM middleware differ depending on the model of ATM, will be described. In this case, designing ATM middleware itself to match the common interface decreases the significance of

20   establishing the common interface.

Therefore a conventional ATM middleware is customized so that the conventional ATM middleware for operating and I/O unit can be utilized and be operated with a common interface.

Fig. 14 and Fig. 15 are diagrams depicting the operation

25   of an embodiment of the present invention, Fig. 16 is a table showing the common interfaces thereof and the vender specific interfaces, and Fig. 17 and Fig. 18 are flow charts depicting

the interface conversion processing in the I/O control
library.

In Fig. 14 and Fig. 15, the ATM application of the
server 100 is a Web application, and the browser 120 issues
5   the CRW (card unit) command and the BRU (bill unit) command
with parameters a and b, and A and B as a common interface in
the ATM applet function call-up format of Java (registered
trademark).

On the other hand, in the ATM 1 as described in relation
10 to Fig. 5, the parameter file 160 stores the input
parameters/fixed parameters which are uniquely determined by
the system specifications of each vender. In Fig. 14, the
specific parameters c and d are stored for the CRW command A,
and the specific parameters C, D and E are stored for the BRU
15 command X as the setup information specific to the company A.

In the same way, in Fig. 15, the specific parameter C is
stored for the CRW command A, and the specific parameters C,
D, E and F are stored for the BRU command X as the setup
information specific to the company B.

20   The I/O control layer 170 calls up the parameter
specific to each I/O client layer 190 from this parameter
file 160 when the ATM middleware 180, including the I/O
client layer 190, is called up, and converts the common API
into the convention client API.

25   For example, in Fig. 14, the I/O control layer 170
converts the CRW command A (a, b) of the common interface
(API) into the command A (a, b, c, d) of the CRW middle API,

32

sends it to the CRW middleware 180, and operates the CRW unit 10, and also converts the BRU command X (A, B) of the common interface (API) into the command X (A, B, C, D, E) of the BRU middle API, sends it to the BRU middleware 180, and operates

5   the BRU unit 50.

        In the same way, in Fig. 15, the I/O control layer 170 converts the CRW command A (a, b) of the common interface (API) into the command A (a, b, c) of the CRW middle API, sends it to the CRW middleware 180, and operates the CRW unit

10 10, and also converts the BRU command X (A, B) of the common interface (API) into the command X (A, B, C, D, E, F) of the BRU middle API, sends it to the BRU middleware 180, and operates the BRU unit 50.

        This will be described using a specific example of the

15 pass book processing command in Fig. 16. For the unit initialization command, the vender specific interfaces (parameters) to be provided are the pass book printer user type code, pass book ribbon near end check availability specification, pass book MS (Magnetic Stripe) error pause

20 availability specification, pass book page markup and down use availability specification and page closing availability specification when a forgotten pass book is taken in, which can be specified depending on the specifications of the pass book unit of each vender.

25   For the pass book insertion processing command, the common interface (parameters) to be provided are the insertion medium logic type specification, pass book type

33

number specification, W-MS (Wide Magnetic Stripe) switching

function, stripe position specification, MS erase

availability specification, specified line position

specification, composite operation specification and camera

5 control (camera is operated when a pass book is inserted)

specification.

The vender specific interfaces (parameters) to be

provided, on the other hand, are the MS position

specification (e.g. the case when the position of the MS of

10 the pass book is different depending on the financial

institution), MS type (recording format), MS parameter (e.g.

whether MS data is erasable) and line lamp specification

(when a line lamp exists at the pass book entry, the lamp is

turned ON at insertion and at ejection).

15   Fig. 17 is a flow chart depicting the above mentioned

unit initialization processing.

(S10) When the unit initialization command is received,

the vender specific setup information (parameters) on

initialization is read from the parameter file 160.

20   (S12) The read parameters are attached to the

initialization command, are then sent to the unit, and a

reply is received.

Fig. 18 is a flow chart depicting the above mentioned

pass book insertion processing.

25   (S20) When the pass book insertion command is received,

the vender specific setup information (parameters) on the

pass book insertion is read from the parameter file 160.

(S22) The parameters of the vender specific unit are
edited from the read parameters and the parameters received
from the common interface (input information).

(S24) The edited parameters are attached to the pass
5 book insertion command, are then sent to the unit, and a
reply is received.

In this way, the vender specific parameters are set in
the parameter file 160 in advance, and the I/O control layer
170 edits the parameters of the command of the common
10 interface (API) and the vender specific parameters of the
parameter file 160, converts it into the command of the ATM
middle API, sends it to the ATM middleware, and operates the
vender specific I/O unit 10, so the vender specific I/O unit
can be operated by the command of the common interface (API).
15 In other words, conventional ATM middleware can be customized
so as to operate with a standard interface.

[Other embodiments]

Now other embodiments of the above mentioned Agent and
method will be described. Fig. 19 is a table showing agents
20 according to another embodiment of the present invention. As
Fig. 19 shows, according to this example, the synchronization
Agent and the method (program) are combined, and the
synchronization Agent has initialization of synchronization
control, mechanism reset, bill/coin insertion, medium
25 simultaneous ejection, printing/feeding/MS write/ejection
preparation, deposit return, storing, forced
ejection/capturing, unit information acquisition/transaction

status setting/two-screen display control, deposit/withdrawal preparation, forced replenishment, jam reset and card/pass book insertion.

The combined Agent/method issues a command to the I/O
5 controller (that is the I/O unit) indicated by the black dot in Fig. 19, and receives the reply of the command execution result. For example, the initialization Agent/method of synchronization control issues the initialization command to the bill controller 175, coin controller 176, pass book
10 controller 171, card controller 172, receipt controller 174, journal controller 177, transaction controller 178 and ten key controller 173 (see Fig. 5), has each controller execute initialization processing, and receives the initialization processing result from each controller as the reply.

15 In the same way, the mechanism reset Agent/method of synchronization control issues the mechanism reset command to the bill controller 175, coin controller 176, pass book controller 171, card controller 172, receipt controller 174, journal controller 177 (see Fig. 5), has each controller
20 execute mechanism reset processing, and receives the mechanism reset processing result from each controller as the reply.

In the same way, the bill/coin insertion Agent/method of synchronization control issues the insertion command to the
25 bill controller 175 and coin controller 176 (see Fig. 5), has each controller execute insertion processing, and receives the insertion processing result from each controller as the

reply.

Fig. 20 shows the screen content to be transmitted by the Web server 100 according to another embodiment of the present invention.

5   As Fig. 20 shows, the Agent name to be called up in the screen is specified in the screen content described in HTML (page description language). As an example, the initialization Agent of the synchronization control (Sync) is called up here by the applet tag <APPLET CODE =

10 "U_agtSync_initial.class".

Also by the description <SCRIPT Language = "javaScript>, call-up of the initialization Agent of the synchronization control is specified using Java script as the method. In other words, call-up of the initialization Agent/method of

15 the synchronization control is specified by the description ret = document.U_agtSync_initial. Here the details of the screen display content are omitted.

When this screen content is downloaded to the browser 120, the initialization Agent of the synchronization control

20 is specified by the applet tag, and the initialization Agent/method of the synchronization control is called up by the call-up method name of the SCRIPT.

As described above, the called up initialization method issues the initialization command to the bill controller 175,

25 coin controller 176, pass book controller 171, card controller 172, receipt controller 174, journal controller 177, transaction controller 178 and ten key controller 173

(see Fig. 5), has each controller execute initialization

processing, and receives the initialization processing result

from each controller as the reply.

   In this case as well, as Fig. 12 shows, the called up

5   initialization method continuously issues the initialization

command to the bill controller 175, coin controller 176, pass

book controller 171, card controller 172, receipt controller

174, journal controller 177, transaction controller 178 and

ten key controller 173 (see Fig. 5), and receives the reply

10   from these controllers sequentially as the processing

   completes.

   Therefore the I/O units can be controlled in parallel,

and even a plurality of I/O units can be controlled in a

short time. Whereas in the case of the method of specifying

15   a method for each I/O unit using conventional Script, which

is a sequential control for issuing commands and receives the

reply sequentially for each script, as shown in Fig. 23, it

takes time to control an I/O by the Web.

   Also the above mentioned Agent is provided for each

20   processing to which a processing unit name is assigned, so

ATMs with various functions can be supported with the same

applet tags and method names by slightly changing the content

of the processing unit. In this example, the content of the

processing unit is changed by changing the initial (input

25   Param) in the parenthesis of the method name of the above

mentioned method callup SCRIPT. In other words, as Fig. 11

shows, the input Param is comprised of the 8-bit information

38

for setting the input flags for each I/O controller (bill, coin, pass book, card, receipt, journal, transaction control and ten key).

The input flag indicates that the input of the method to the I/O controller where "1" is set is enabled. The called up method refers to this input flag, and determines an I/O controller to which the command is issued. Therefore even ATMs with different functions can execute a same processing with a same applet tag and method name by operating the input flag by the Web server 100.

For example, if the input flag of coins is set to "0" in a transaction apparatus which does not handle coin, issuing a command to the coin controller can be prevented. In the same way, if the pass book input flag is set to "0" in a transaction apparatus which does not handle pass books, issuing a command to the pass book controller can be prevented.

According to this embodiment, compared with the example in Fig. 7, the applet (Agent) to be sent with the screen content can be divided into relatively small capacity programs, so the data volume of transmission content can be decreased, and communication time can be decreased.

In the case of the example of the Agent in Fig. 7, the applet of synchronization control is a relatively large capacity program, because the applet has many methods, but by saving the applet to the cache in the ATM 1, the transmission of this applet becomes unnecessary from the next time it is

called up, and the data volume of the transmission content from the next time can be decreased, and communication time can be decreased.

In the above mentioned embodiments, the automatic

5   deposit/withdrawal machine was described as an example of the automatic transaction apparatus shown in Fig. 1, but the present invention can be applied to other apparatus, such as a withdrawal machine, automatic cash loan machine and automatic issuing machine. The network was described using

10   the Internet, but the present invention can also be applied to other networks, and the server can be applied not only for Java SCRIPT, but for other SCRIPTS as well.

The present invention was also described using customized middleware control by the common API as an example,

15   but the present invention can be applied to middleware control without customization by the common API.

The present invention was described using the embodiments, but the present invention can be modified in various ways within the scope of the essential character of

20   the present invention, which shall not be excluded from the scope of the present invention.

According to the present invention, operation of a plurality of I/O units can be controlled by embedding an Agent for each processing of transaction processing in the

25   screen content of the Web server, and calling up one method for each processing, and the plurality of I/O units can be commonly used by different automatic transaction apparatus in

a general transaction processing flow. Therefore the screen
content for Web-controlled automatic transaction apparatus
with different functions and configurations can be easily
created, parallel control becomes possible, and high-speed

5   I/O unit control can be implemented. Therefore the wait time
of the user for the automatic transaction apparatus can be
decreased, and the operation rate can be improved, which
contributes to the popularization of automatic transaction
apparatus by Web control.

10